

## Module:5 sql database

### 1. Create Table Name : Student and Exam

- **Create table query:**

- Create table student

```
(  
    Rollno int primary key,  
    Name varchar(15),  
    Branch varchar(15)  
);
```

- Create table exam

```
(  
    Rollno int,  
    S_code varchar(10),  
    Marks int,  
    P_code varchar(10),  
  
    FOREIGN KEY(rollno) REFERENCES    student(rollno)  
);
```

- **Insert query:**

```
insert into student values(1,'jayesh','computer science');  
insert into student values(2,'jay','electric and com');  
insert into student values(3,'jignesh',' electric and com');
```

- **Output:(student table) and (exam table)**

▼	rollno	name	branch
	1	jayesh	computer science
	2	jay	electric and com
	3	jignesh	electric and com

## Module:5 sql database

rollno	s_code	marks	p_code
1	cs11	50	cs
1	cs12	60	cs
2	EC101	54	ec
2	EC102	48	ec
3	EC101	60	ec
3	EC102	58	ec

### 2. Create table : Employee and IncentiveTable

- **Create table query:**

- Create table employee

```
(  
    Emp_id int primary key,  
    First_Name varchar(15),  
    last_Name varchar(15),  
    salary int,  
    joining_date varchar(15),  
    department varchar(15)  
);
```

- Create table incentive

```
(  
    Employee_id int,  
    Incentive_date varchar(10),  
    Incentive_amount int,  
  
    FOREIGN KEY(employee_id) REFERENCES employee(employee_id)  
);
```

- **Insert query:(employee)**

```
insert into employee values(1,'samat' ,modhavadiya',200000,'1-FEB-12 12:00:00  
AM','service');
```

```
insert into employee values(2,karan ,modhavadiya',200000,'1-FEB-12 12:00:00  
AM','service');
```

## Module:5 sql database

```
insert into employee values(3,sameer ,'modhavadiya',200000,'1-FEB-12 12:00:00 AM','service');
```

```
insert into employee values(4,ajay ,'modhavadiya',200000,'1-FEB-12 12:00:00 AM','service');
```

```
insert into employee values(5,raj ,'modhavadiya',200000,'1-FEB-12 12:00:00 AM','service');
```

- **Insert query:(incentive)**

```
insert into incentive values(1,'1-FEB-13',5000);
```

```
insert into incentive values(2,'1-FEB-13',4000);
```

```
insert into incentive values(3,'1-JAN-13',3500);
```

```
insert into incentive values(1,'1-JAN-13',3000);
```

```
insert into incentive values(2,'1-JAN-13',2500);
```

- **Output:(employee)**

employee_id	first_name	last_name	salary	joining_date	department
5	sameer	modhavadiya	180000	01-FEB-13 01.00.00 A	service
1	samat	modhavadiya	300000	01-JAN-13 12.00.00 A	sales
3	karan	modhavadiya	200000	01-FEB-13 01.00.00 A	sales
6	sarman	odedara	225000	02-FEB-13 12.00.00 A	sales
2	ajay	odedara	230000	01-JAN-13 12.00.00 A	insurance
4	gaurav	sisodiya	400000	01-FEB-13 01.00.00 A	banking

- **Output:(incentive)**

emp_id	incentive_date	incentive_amount
1	01-FEB-13	5000
2	01-FEB-13	4000
3	01-JAN-13	3500
1	01-JAN-13	3000
2	01-JAN-13	2500

3. Get First\_Name from employee table using name "Employee Name".

- **Query:**

```
SELECT First_Name FROM Employee WHERE First_Name ='karan';
```

- **Output:**

## Module:5 sql database

First_Name
karan

4. Get FIRST\_NAME, Joining Date, and Salary from employee table.

- **Query:**

SELECT FIRST\_NAME, Joining\_Date, Salary FROM Employee;

- **Output:**

first_name	joining_date	salary
samat	01-JAN-13 12.00.00 A	300000
ajay	01-JAN-13 12.00.00 A	230000
karan	01-FEB-13 01.00.00 A	200000
gaurav	01-FEB-13 01.00.00 A	400000
sameer	01-FEB-13 01.00.00 A	180000
sarman	02-FEB-13 12.00.00 A	225000

5. Get all employee details from the employee table order by First\_Name Ascending and Salary descending?

- **Query:**

SELECT \* FROM Employee ORDER BY First\_Name ASC, Salary DESC;

- **Output:**

employee_id	first_name	last_name	salary	joining_date	department
2	ajay	odedara	230000	01-JAN-13 12.00.00 A	insurance
4	gaurav	sisodiya	400000	01-FEB-13 01.00.00 A	banking
3	karan	modhavadiya	200000	01-FEB-13 01.00.00 A	sales
1	samat	modhavadiya	300000	01-JAN-13 12.00.00 A	sales
5	sameer	modhavadiya	180000	01-FEB-13 01.00.00 A	service
6	sarman	odedara	225000	02-FEB-13 12.00.00 A	sales

## Module:5 sql database

6. Get employee details from employee table whose first name contains 'J'.

- **Query:**

```
SELECT * FROM Employee WHERE First_Name LIKE '%s%';
```

- **Output:**

employee_id	first_name	last_name	salary	joining_date	department
1	samat	modhavadiya	300000	01-JAN-13 12.00.00 A	sales
5	sameer	modhavadiya	180000	01-FEB-13 01.00.00 A	service
6	sarman	odedara	225000	02-FEB-13 12.00.00 A	sales

7. Get department wise maximum salary from employee table order by salary ascending?

- **Query:**

```
SELECT Department, MAX(Salary) AS Max_Salary FROM Employee GROUP BY  
Department ORDER BY Max_Salary ASC;
```

- **Output:**

department	Max_Salary
service	180000
insurance	230000
sales	300000
banking	400000

9. Select first\_name, incentive amount from employee and incentives table for those employees who have incentives and incentive amount greater than 3000.

- **Query:**

## Module:5 sql database

```
SELECT e.First_Name, i.Incentive_Amount FROM Employee e JOIN Incentive i ON  
e.employee_id = i.emp_id WHERE i.Incentive_amount > 3000;
```

- **Output:**

first_Name	incentive_Amount
samat	5000
ajay	4000
karan	3500

10. Create After Insert trigger on Employee table which insert records in viewtable.

- **Create table query:**

- Create table viewtable

```
(  
    e_id int primary key,  
    f_name varchar(15),  
    l_name varchar(15),  
    e_salary int,  
    j_date varchar(15),  
    e_department varchar(15)  
);
```

- **Query:**

```
CREATE TRIGGER tri_viewtable AFTER INSERT ON Employee  
FOR EACH ROW  
BEGIN  
    INSERT INTO viewtable (employee_id, first_name, last_name, salary, joining_date,  
department) VALUES (NEW.e_id, NEW.f_Name, NEW.l_Name, NEW.e_salary, NEW.j_Date,  
NEW.e_Department);  
END;
```

- **Insert query(employee):**

```
insert into employee values(6,sarman,'odedara',225000,'12-FEB-13 12:00:00 AM',  
'service');
```

- **Output:(in viewtable)**

## Module:5 sql database

e_id	f_name	i_name	e_salary	i_date	e_department	actionperformed
6	sarman	odedara	225000	02-FEB-13 12:00:00 A	sales	Record inserted

☐ Show all | Number of rows: 25 ▼ | Filter rows:

11.Create table given below: Salesperson and Customer

- **Create table query:**

- Create table salesperson

```
(  
    S_no int primary key,  
    S_name varchar(15),  
    S_city varchar(15)  
    S_commission int  
);
```

- Create table customer

```
(  
    C_no int,  
    C_name varchar(10),  
    C_city varchar(10),  
    Rating int,  
    S_no int,
```

```
        FOREIGN KEY(s_no) REFERENCES      salesperson (s_no)  
);
```

- **Insert query:(salesperson)**

```
insert into salesperson values(1001,'peel' ,london' ,12);
```

```
insert into salesperson values(1002,'serres' ,san jose',13);
```

```
insert into salesperson values(1003,'motika' ,london' ,11);
```

```
insert into salesperson values(1004,'rafkin' ,barcelona' ,15);
```

```
insert into salesperson values(1005,'axeirod' ,new york' ,1);
```

## Module:5 sql database

- **Insert query:(customer)**

insert into customer values(201,'hoffman','london',100,1001);

insert into customer values(202,'giovanne','roe',200,1003);

insert into customer values(203,'liu','san jose',300,1002);

insert into customer values(204,'grass','barcelona',100,1002);

insert into customer values(205,'simon','london',300,1004);

- **Output:(salesperson)**

s_no	s_name	city	commision
1001	peel	london	12
1002	serres	san jose	13
1003	motika	london	11
1004	rafkin	barcelona	15
1005	axelrod	new york	1

- **Output:(customer)**

c_no	c_name	c_city	rating	s_no
201	hoffman	london	100	1001
202	giovanne	roe	200	1003
203	liu	san jose	300	1002
204	grass	barcelona	100	1002
205	simon	london	300	1004

13.All orders for more than \$100.

- **Query:**

SELECT \* FROM customer WHERE rating > 100;

- **Output:(customer)**

c_no	c_name	c_city	rating	s_no
202	giovanne	roe	200	1003
203	liu	san jose	300	1002
205	simon	london	300	1004

14. Names and cities of all salespeople in London with commission above 0.12

- **Query:**

SELECT S\_name, city FROM salesperson WHERE s\_city = 'london' AND commission >10;



## Module:5 sql database

- **Output:(salesperson)**

S_name	city
peel	london
motika	london

15. All salespeople either in Barcelona or in London.

- **Query:**  
SELECT \* FROM salesperson WHERE City = 'Barcelona' OR City = 'London';
- **Output:(salesperson)**

s_no	s_name	city	commission
1001	peel	london	12
1003	motika	london	11
1004	rafkin	barcelona	15

16.All salespeople with commission between 0.10 and 0.12.(Boundary values should be excluded).

- **Query:**  
SELECT \* FROM salesperson WHERE commission > 10 AND commission < 15;

- **Output:(salesperson)**

s_no	s_name	city	commission
1001	peel	london	12
1002	serres	san jose	13
1003	motika	london	11

17.All customers excluding those with rating <= 100 unless they are located in London.

- **Query:**  
SELECT \* FROM customer WHERE Rating > 100 OR c\_city = 'london';
- **Output:(customer)**

## Module:5 sql database

c_no	c_name	c_city	rating	s_no
201	hoffman	london	100	1001
202	giovanne	roe	200	1003
203	liu	san jose	300	1002
205	simon	london	300	1004

18. Write a SQL statement that displays all the information about all salesperson.

- **Query:**

```
SELECT * FROM salesperson;
```

- **Output:(salesperson)**

s_no	s_name	city	commision
1001	peel	london	12
1002	serres	san jose	13
1003	motika	london	11
1004	rafkin	barcelona	15
1005	axelrod	new york	1

19. From the following table, write a SQL query to find orders that are delivered by a salesperson with ID. 5001. Return ord\_no, ord\_date, purch\_amt.

- **Create table query:(orders)**

- Create table orders

```
(  
    Order_no int PRIMARY KEY,  
    Pur_amount int,  
    Order_date varchar(10),  
    C_no int,  
    S_no int,  
  
    FOREIGN KEY(c_no) references customer(c_no),  
    FOREIGN KEY(s_no) references salesperson(s_no)  
);
```

- **Insert query:**

```
insert into orders values(70001,150,'2012-10-05',201,1002);
```

```
insert into orders values(70003,300,'2012-11-02',202,1005);
```

```
insert into orders values(70006,502,'2012-11-18',201,1003);
```

```
insert into orders values(70007,205,'2012-11-10',205,1004);
```

```
insert into orders values(70009,1500,'2012-10-09',204,1004);
```

- **Output:(orders)**

order_no	pur_amount	order_date	c_no	s_no
70001	150	2012-10-05	201	1002
70003	300	2012-11-02	202	1005
70006	502	2012-11-18	201	1003
70007	205	2012-11-10	205	1004
70009	1500	2012-10-09	204	1004

- **Query:**

```
SELECT order_no, order_date, pur_amount FROM orders WHERE s_no = 1004;
```

- **Output:(orders)**

## Module:5 sql database

order_no	order_date	pur_amount
70007	2012-11-10	205
70009	2012-10-09	1500

20. From the following table, write a SQL query to select a range of products whose price is in the range Rs.200 to Rs.600. Begin and end values are included. Return pro\_id, pro\_name, pro\_price, and pro\_com.

- **Create table query:(item\_mast)**
- Create table item\_mast  
(  
    Pro\_id int PRIMARY KEY,  
    Pro\_name varchar(20),  
    Pro\_prize int,  
    Pro\_com int,  
);
- **Insert query:**  
  
    insert into orders values(101,'mother board',3200,15);  
  
    insert into orders values(101,'mother board',3200,15);  
  
    insert into orders values(101,'mother board',3200,15);  
  
    insert into orders values(101,'mother board',3200,15);  
  
    insert into orders values(101,'mother board',3200,15);

- **Output:(item\_mast)**

pro_id	pro_name	pro_price	pro_com
101	mother board	3200	15
102	keyboard	450	16
103	zip drive	250	14
104	speaker	550	16
105	moniter	5000	11

- **Query:**  
  
    SELECT pro\_id, pro\_name, pro\_price, pro\_com FROM item\_mast WHERE pro\_price  
    BETWEEN 200 AND 600;

## Module:5 sql database

- **Output:(item\_mast)**

pro_id	pro_name	pro_price	pro_com
102	keyboard	450	16
103	zip drive	250	14
104	speaker	550	16

21.From the following table, write a SQL query to calculate the average price for a manufacturer code of 16 Return avg.

- **Query:**

```
SELECT AVG(pro_price) AS avg FROM item_mast WHERE pro_com = 16;
```

- **Output:(item\_mast)**

avg
500.0000

22. From the following table, write a SQL query to display the pro\_name as 'Item Name' and pro\_priceas 'Price in Rs.'

- **Query:**

```
SELECT pro_name AS 'Item Name', pro_price AS 'Price in Rs.' FROM item_mast;
```

- **Output:(item\_mast)**

Item Name	Price in Rs.
mother board	3200
keyboard	450
zip drive	250
speaker	550
moniter	5000

23. From the following table, write a SQL query to find the items whose prices are higher than or equal to \$250. Order the result by product price in descending, then product name in ascending. Return pro\_name and pro\_price.

- **Query:**

## Module:5 sql database

```
SELECT pro_name, pro_price FROM item_mast WHERE pro_price >= 250 ORDER BY  
pro_price DESC, pro_name ASC;
```

○ **Output:(item\_mast)**

pro_name	pro_price
moniter	5000
mother board	3200
speaker	550
keyboard	450
zip drive	250

24. From the following table, write a SQL query to calculate average price of the items for each company. Return average price and company code.

○ **Query:**

```
SELECT pro_com, AVG(pro_price) AS average_price FROM item_mast GROUP BY pro_com;
```

○ **Output:(item\_mast)**

pro_com	average_price
11	5000.0000
14	250.0000
15	3200.0000
16	500.0000